A Generic Sensor Fusion Problem: Classification and Function Estimation

Nageswara S. V. Rao[1]

Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA,
raons@ornl.gov

Abstract. A generic fusion problem is studied for multiple sensors whose outputs are probabilistically related to their inputs according to unknown distributions. Sensor measurements are provided as iid input-output samples, and an empirical risk minimization method is described for designing fusers with distribution-free performance bounds. The special cases of isolation and projective fusers for classifiers and function estimators, respectively, are described in terms of performance bounds. The isolation fusers for classifiers are probabilistically guaranteed to perform at least as good as the best classifier. The projective fusers for function estimators are probabilistically guaranteed to perform at least as good as the best subset of estimators.

1 Introduction

The information fusion problems have been solved for centuries in various disciplines, such as political economy, reliability, pattern recognition, forecasting, and distributed detection. In multiple sensor systems, the fusion problems arise naturally when overlapping regions are covered by the sensors. Often, the individual sensors can themselves be complex, consisting of sophisticated sensor hardware and software. Consequently, sensor outputs can be related to the actual object features in a complicated manner, and these relationships are often characterized by probability distributions. Early information fusion methods required statistical independence of sensor errors, which greatly simplified the fuser design; for example, a weighted majority rule suffices in detection problems. Such solutions are not applicable to current multiple sensor systems, since the sensors measurements can be highly correlated and consequently violate the statistical independence property. Another classical approach to fuser design is the Bayesian method that minimizes a suitable expected risk, which relies on analytical expressions for sensor distributions. Deriving the required closed-form sensor distributions is very difficult since it often requires the knowledge of areas such as device physics, electrical engineering, and statistical modeling. Particularly when only a finite number of measurements are available, the selection of a fuser from a carefully chosen function class is easier, in a fundamental information-theoretic sense, than inferring completely unknown sensor distributions [21].

In operational sensor systems measurements are collected by sensing objects and environments with known parameters. Thus fusion methods that utilize such empirical observational or experimental data will be of high practical relevance. In this paper, we present a brief overview of rigorous approaches for designing such fusers based on the empirical process theory [21] to provide performance guarantees based on finite samples. We briefly describe a general fuser design approach and illustrate it using a vector space method. A more detailed account of the generic sensor fusion problem can be found in [18]. The problem of combining outputs of multiple classifiers is a special case of the generic sensor fusion problem, wherein the training sample corresponds to the measurements. We describe the isolation fuser methods for classifiers to probabilistically ensure that fuser's performance guarantees are at least as good as those of best classifier. The fusion of function estimators is another special case of the sensor fusion problem which is of practical utility. We then describe the nearest-neighbor projective fuser for function estimators that performs at least as good as the best projective combination of the estimators. Both isolation and projective fusers have been originally developed for the generic sensor fusion problem, and we sharpen the general performance results for these special cases.

This paper presents a brief account of results from other papers. We describe the classical sensor fusion methods in Section 2. We present a generic sensor fusion problem and a solution using empirical risk minimization in Section 3. We describe the problem of fusing classifiers and function estimators in Sections 4 and 5, respectively. The original notations from the respective areas are retained in the individual sections; while it results in a non-uniform notation, it makes it easier to relate these results to the individual areas.

2 Classical Fusion Problems

Fusion methods for multiple sources to achieve performances exceeding those of individual sources have been studied in political economy models in 1786 and composite methods in 1818. In the twentieth century, fusion methods have been applied in a wide spectrum of areas such as reliability, forecasting, pattern recognition, neural networks, decision fusion, and statistical estimation. A brief overview of early information fusion works can be found in [7]. The problem of fusing classifiers is relatively new and is first addressed in a probabilistic framework by Chow [1] in 1965.

When sensor distributions are known, several fusion rule estimation problems have been solved under various formulations. A simpler version of this problem is the Condorcet jury model (see [4] for an overview), where a majority rule can be used to combine 1-0 probabilistically independent decisions of a group of N members. If each member has probability p of making a correct decision, the probability that the majority makes the correct decision is $p_N = \sum\limits_{i=N/2}^{N} \binom{N}{i} p^i (1-p)^{N-i}$. Then we have an interesting dichotomy: (a) if $p > 0.5$, then $p_N > p$ and $p_N \to 1$ as $N \to \infty$; and (b) if $p < 0.5$, then $p_N < p$ and $p_N \to 0$ as $N \to \infty$.

For the boundary case $p = 0.5$ we have $p_N = 0.5$. Interestingly, this result has been rediscovered by von Neumann in 1959 in building reliable computing devices using unreliable components by taking a majority vote of duplicated components. For multiple classifiers, a weighted majority fuser is optimal [1] under statistical independence, and the fuser weights can be derived in a closed-form using the classifier detection probabilities. Over the past few years, multiple classifier systems have witnessed an extensive interest and growth [5, 23].

The distributed detection problem [22] studied extensively in the target tracking area can be viewed as a generalization of the above two problems. The Boolean decisions from a system of detectors are combined by minimizing a suitably formulated Bayesian risk function. The risk function is derived from the detector densities and the minimization is typically carried out using analytical or deterministic optimization methods. In particular, the risk function used for classifier fusion in [1] corresponds to the misclassification probability and its minima is achieved by the weighted majority rule. In these works, the sensor distributions are assumed to be known, which is reasonable in their domains. While several of these solutions can be converted into sample-based ones [9], these are not primarily designed for measurements. As evidenced in practical multiple sensor systems and classifiers, it is more pragmatic to have the measurements rather than the error distributions.

3 A Generic Sensor Fusion Problem

We consider a multiple sensor system of N sensors, where sensor S_i, $i = 1, 2, \ldots, N$, outputs $Y^{(i)} \in \Re^d$ corresponding to input $X \in \Re^d$ according to distribution $P_{Y^{(i)}|X}$. Intuitively, input X is the "measured" quantity such as presence of a target or a value of feature vector. The *expected error* of sensor S_i is defined as

$$I(S_i) = \int C\left(X, Y^{(i)}\right) dP_{Y^{(i)},X},$$

where $C : \Re^d \times \Re^d \mapsto \Re$ is the cost function. Here, $I(S_i)$ is a measure of how good sensor S_i is in "measuring" input feature X. If S_i is a detector [22] or classifier [2], we can have $X \in \{0, 1\}$ and $Y^{(i)} \in \{0, 1\}$, where $X = 1$ (0) corresponds to presence (absence) of a target. Then $I(S_i) = \int \left[X \oplus Y^{(i)}\right] dP_{Y^{(i)},X}$ is the probability of misclassification (false alarm and missed detection) of S_i, where $\oplus$ is the exclusive-OR operation [1] .

The *measurement error* corresponds to the randomness in measuring a particular value of feature X, which is distributed according to $P_{Y^{(i)}|X}$. The *systematic error* at X corresponds to $E[C(X, Y^{(i)})|X]$ which must be 0 in the case of a perfect sensor. This error is often referred to as the bias error.

We consider a fuser $f : \Re^{Nd} \mapsto \Re^d$ that combines the outputs of sensors $Y = \left(Y^{(1)}, Y^{(2)}, \ldots, Y^{(N)}\right)$ to produce the fused output $f(Y)$. We define the

[1] Alternatively, X can be expanded to include the usual "feature" vector and $C(.)$ can be redefined so that $I(S_i)$ is misclassification probability.

expected error of the fuser f to be

$$I_F(f) = \int C(X, f(Y))dP_{Y,X}$$

where $Y = \left(Y^{(1)}, Y^{(2)}, \ldots, Y^{(N)}\right)$. The objective of fusion is to achieve low values of $I_F(f)$, and for this both systematic and measurement errors must be taken into account. The fuser is typically chosen from a family of fusion rules $\mathcal{F} = \{f : \Re^{Nd} \mapsto \Re^d\}$ which could be either explicitly or implicitly identified. The *expected best* fusion rule f^* satisfies $I_F(f^*) = \min_{f \in \mathcal{F}} I_F(f)$. For example, if $\mathcal{F}$ is a set of sigmoidal neural networks obtained by varying the weight vector for a fixed architecture, then $f^* = f_{w^*}$ corresponds to the weight vector w^* that minimizes $I_F(.)$ over all weight vectors.

In this formulation, since $I_F(.)$ depends on $P_{Y,X}$, f^* cannot be computed even in principle if the distribution is not known. We consider that only an independently and identically distributed (iid) l-sample (X_1, Y_1), (X_2, Y_2), $\ldots$, (X_l, Y_l) is given, where $Y_i = \left(Y_i^{(1)}, Y_i^{(2)}, \ldots, Y_i^{(N)}\right)$ and $Y_i^{(j)}$ is the output of S_j in response to input X_i. Our goal is to obtain an estimator $\hat{f}$, based *only* on a sufficiently large sample, such that

$$P^l_{Y,X}\left[I_F(\hat{f}) - I_F(f^*) > \epsilon\right] < \delta \qquad (1)$$

where $\epsilon > 0$ and $0 < \delta < 1$, and $P^l_{Y,X}$ is the distribution of iid l-samples. As per this condition the "error" of $\hat{f}$ is within ϵ of optimal error (of f^*) with probability $1 - \delta$, *irrespective* of the sensor distributions. Since $\hat{f}$ is to be "chosen" from a potentially infinite set, namely $\mathcal{F}$, based only on a finite sample, this condition is a reasonable target. Strictly stronger conditions are generally not possible to achieve. For example, consider the condition $P^l_{Y,X}[I_F(\hat{f}) > \epsilon] < \delta$ for the case case of classifiers $\mathcal{F} = \{f : [0,1]^N \mapsto \{0,1\}\}$. This condition cannot be satisfied, since for any classifier $f \in \mathcal{F}$, there exists a distribution for which $I_F(f) > 1/2 - \rho$ for any $\rho \in [0,1]$ (see Theorem 7.1 of [2] for details).

Consider a simple two-sensor system such that $Y^{(1)} = a_1 X + Z$, where Z is normally distributed with zero mean, and is independent of X, i. e. a constant scaling error and a random additive error. For the second sensor, we have $Y^{(2)} = a_2 X + b_2$, which has a scaling and bias error. Let X be uniformly distributed over $[0,1]$, and $C[X,Y] = (X - Y)^2$. Then, we have $I(S_1) = (1 - a_1)^2$ and $I(S_2) = (1 - a_2 - b_2)^2$, which are non zero in general. For

$$f\left(Y^{(1)}, Y^{(2)}\right) = \frac{Y^{(1)}}{2a_1} + \frac{1}{2a_2}(Y^{(2)} - b_2).$$

we have $I_F(f) = 0$, since the bias b_2 is subtracted from $Y^{(2)}$ and the multipliers cancel the scaling error. Such fuser can be designed only with a significant insight into sensors, in particular with a detailed knowledge about the distributions. To illustrate the effects of finite samples, we generate three values for X given by

$\{0.1, 0.5, 0.9\}$ with corresponding Z values given by $\{0.1, -0.1, -0.3\}$. The corresponding values for $Y^{(1)}$ and $Y^{(2)}$ are given by $\{0.1a_1+0.1, 0.5a_1-0.1, 0.9a_1-0.3\}$ and $\{0.1a_2 + b_2, 0.5a_2 + b_2, 0.9a_2 + b_2\}$ respectively. Consider a linear fuser $f\left(Y^{(1)}, Y^{(2)}\right) = w_1 Y^{(1)} + w_2 Y^{(2)} + w_3$. The following weights enable the fuser outputs to exactly match X values for each measurement:

$$w_1 = \frac{1}{0.2 - 0.4a_1}, w_2 = \frac{1}{0.4a_2} \quad \text{and} \quad w_3 = \frac{0.1a_1 + 0.1}{0.4a_1 + 0.1} - \frac{0.1a_2 + b_2}{0.4a_2}.$$

While these weights achieve zero error on the measurements they do not achieve zero value for I_F (even though a fuser with zero expected error exists and can be computed if the sensor distributions are given). The idea behind the criterion in Eq 1 is to achieve performances close to optimal using only a sample. To achieve this a suitable $\mathcal{F}$ is selected first, from which a fuser is chosen to achieve small error on a sufficiently large sample, as will be illustrated subsequently.

Due to the generic nature of the sensor fusion problem described here, it is related to a number of similar problems in a wide variety of areas. A detailed discussion of these aspects can be found in [18].

3.1 Empirical Risk Minimization

Consider that the *empirical error estimate*

$$I_{emp}(f) = \frac{1}{l} \sum_{i=1}^{l} \left[X_i - f\left(Y_i^{(1)}, Y_i^{(2)}, \ldots, Y_i^{(N)} \right) \right]^2$$

is minimized by $\hat{f} \in \mathcal{F}$. Such a method corresponds to an ad hoc approach of choosing a class of fusers such as neural networks or linear fusers, and choosing a particular fuser to minimize the error within the class. Performance of such method, including the basic feasibility, depends on the fuser class and the complexity of minimizing the empirical error. For example, if $\mathcal{F}$ has finite capacity [21], then under bounded error, or bounded relative error for sufficiently large sample, we have $P^l_{Y,X}\left[I_F(\hat{f}) - I_F(f^*) > \epsilon \right] < \delta$ for arbitrarily specified $\epsilon > 0$ and δ, $0 < \delta < 1$. Typically, the required sample size is expressed in terms of ϵ and δ and the parameters of $\mathcal{F}$. The most general result [13] that ensures this condition is based on the scale-sensitive dimension, which establishes the basic tractability of this problem. But this general method often results in very loose bounds for the sample size, and tighter estimates are possible by utilizing specific properties of $\mathcal{F}$.

If $\mathcal{F}$ is a vector space of dimensionality d_V, we have the following results[12]: (a) the sample size is a simple function of d_V, (b) $\hat{f}$ can be computed using least square methods in polynomial time, and (c) no smoothness conditions are required on the functions or distributions. For simplicity consider that $X \in [0, 1]$ and $Y \in [0, 1]^N$. Let f^* and $\hat{f}$ denote the expected best and empirical best fusion functions chosen from a vector space $\mathcal{F}$ of dimension d_V and range $[0,1]$. Given an iid sample of size

$$\frac{512}{\epsilon^2}\left[d_V \ln\left(\frac{64e}{\epsilon}+\ln\frac{64e}{\epsilon}\right)+\ln(8/\delta)\right],$$

we have $P\left[I_F(\hat{f})-I_F(f^*)>\epsilon\right]<\delta$ (see [12] for details). This method subsumes two very important cases [12]:

(a) *Potential Functions:* The potential functions where $f_i(y)$ is of the form $exp((y-\alpha)^2/\beta)$ for suitably chosen constants α and β, constitute an example of the vector space method.
(b) *Special Neural Networks:* In two-layer sigmoidal networks of [6], the unknown weights are only in the output layer, which enables us to express each network in the form $\sum\limits_{k=1}^{d_V} a_i\eta_i(y)$ with universal $\eta_i(.)$'s.

Similar sample size estimates have been derived for fusers based on feedforward neural networks in [10]. Also non-linear statistical estimators can be employed to estimate the fuser based on the sample, such as the Nadaraya-Watson estimator [12]. The main limitation of empirical risk minimization approach is that $\hat{f}$ is only guaranteed to be close to f^* but there are no guarantees that the latter is any good. While it is generally true that if $\mathcal{F}$ is large enough, f^* would perform better than best sensor, it is indeed possible that it performs worse than worst sensor. Systematic approaches such as isolation fusers [17] and projective fusers [15] would be useful to ensure the fuser performance. We will subsequently discuss the special cases of isolation and projective fusers for classifiers [11] and function estimators [19], respectively. We note that projective fusers have also been applied to classifiers [11] and isolation fusers have also been applied to function estimators [16].

3.2 Example

We consider 5 classifiers such that $Y \in \{0,1\}^5$ such that $X \in \{0,1\}$ corresponds to "correct" class, which is generated with equal probabilities, i. e., $P(X=0)=P(X=1)=1/2$ [20]. The error of classifier C_i, $i=1,2,\ldots,5$, is described as follows: the output $Y^{(i)}$ is correct decision with probability of $1-i/10$, and is the opposite with probability $i/10$. The task is to combine the outputs of classifiers

Sample Size	Test set	C_1	C_2	C_3	C_4	C_5	Nadaraya-Watson
100	100	7.0	20.0	33.0	35.0	55.0	12.0
1000	1000	11.3	18.5	29.8	38.7	51.6	10.6
10000	10000	9.5	20.1	30.3	39.8	49.6	8.58
50000	50000	10.0	20.1	29.8	39.9	50.1	8.860

Table 1. Percentage error of Nadaraya-Watson estimator and individual classifiers.

Sample Size	Test Size	Bayesian Fuser	Empirical Decision	Nearest Neighbor	Nadaraya-Watson
100	100	91.91	23.00	82.83	88.00
1000	1000	91.99	82.58	90.39	89.40
10000	10000	91.11	90.15	90.81	91.42
50000	50000	91.19	90.99	91.13	91.14

Table 2. Correct classification percentage of fusers.

to predict the correct class. The percentage error of the individual classifiers and the fused system based on the Nadaraya-Watson estimator is presented in Table 1. Note that the fuser is consistently better than the best classifier C_1 beyond the sample size of 1000. The performance results of Nadaraya-Watson estimator, empirical decision rule, nearest neighbor rule, and Bayesian rule based on the analytical formulas are presented in Table 2. The Bayesian rule is computed based on the formulas used in the data generation and is provided for comparison only.

4 Isolation Fusers for Classifiers

Over the past decades several methods, such as nearest neighbor rules, neural networks, tree methods, and kernel rules, have been developed for designing classifiers. Often, the classifiers are quite varied and their performances are characterized by various smoothness and/or combinatorial parameters [2]. The designer is thus faced with a wide variety of choices which are not easily comparable. It is generally known that a good fuser outperforms the best classifier, and at the same time, a bad fuser choice can result in a performance worse than the worst classifier. Thus it is very important to employ fusion methods that provide concrete performance guarantees – in particular, for the fuser to be reasonable it must perform at least as well as the best classifier.

We are given an independently and identically distributed (iid) sample (X_1, Y_1), (X_2, Y_2), …, (X_n, Y_n), according to an unknown distribution $P_{X,Y}$, where $X_i \in \Re^d$ and $Y_i \in \{0, 1\}$. The problem is to design a classifier $\phi : \Re^d \mapsto \{0, 1\}$ based on the sample that ensures a small value for the *probability of misclassification*

$$L(\phi) = \int\limits_X I_{\{\phi(X)\neq Y\}} dP_{X,Y},$$

where $I_D(x)$ is the *indicator function* of the set $D \subseteq \Re^d$ such that $I_C(x) = 1$ if $x \in C$ and $I_C(x) = 0$ otherwise. We often suppress the operand x when it is clear from the context.

For $\phi \in \mathcal{H}$, the *empirical error of misclassification* is given by

$$\hat{L}(\phi) = \frac{1}{n} \sum_{i=1}^{n} I_{\{\phi(X_i)\neq Y_i\}}.$$

Let $\hat{\phi}$ minimize $\hat{L}(.)$ over $\mathcal{H}$. If $\mathcal{H}$ has finite Vapnik-Chervonenkis dimension $V_{\mathcal{H}}$, we have [2]

$$P^n_{X,Y}\left[L(\hat{\phi}) - \min_{\phi\in\mathcal{H}} L(\phi) > \epsilon\right] \leq \delta$$

for sufficiently large n, irrespective of the distribution $P_{X,Y}$. We are given N such classifiers corresponding to the classes $\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_N$ such that

$$P^n_{X,Y}\left[L(\hat{\phi}_i) - \min_{\phi\in\mathcal{H}_i} L(\phi) > \epsilon\right] \leq \delta_i$$

where $\hat{\phi}_i$ minimizes $\hat{L}(.)$ over $\mathcal{H}_i$. Our objective is to "fuse" the classifier outputs so that the fused system performs at least as well as the best individual classifier based on the sample only. We next describe a method based on the isolation property that enables us to compare the fused system with the best individual classifier [11]. This method is simple to apply and requires easily satisfiable criteria.

4.1 Single Classifier

The lowest possible error achievable by any deterministic classifier is given by the *Bayes error* $L(\phi^*)$, where $\phi^* : \Re^d \mapsto \{0, 1\}$ is defined as

$$\phi^*(x) = \begin{cases} 1 \text{ if } P_{X,Y}[Y = 1|X = x] \geq P_{X,Y}[Y = 0|X = x] \\ 0 \text{ otherwise} \end{cases}$$

Since the distribution is not known, ϕ^* cannot be computed. The performance of $\hat{\phi}$ that minimizes $\hat{L}(.)$ can be characterized using the properties of $\mathcal{H}$.

Let $\mathcal{A}$ be a collection of measurable sets of R^d. For $(z_1, z_2, \ldots, z_n) \in \{\Re^d\}^n$, let $\mathcal{N}_{\mathcal{A}}(z_1, z_2, \ldots, z_n)$ denote the number of different sets in $\{\{z_1, z_2, \ldots, z_n\} \cap A : A \in \mathcal{A}\}$. The nth *shatter coefficient* of $\mathcal{A}$ is

$$s(\mathcal{A}, n) = \max_{(z_1,z_2,\ldots,z_n)\in\{\Re^d\}^n} \mathcal{N}_{\mathcal{A}}(z_1, z_2, \ldots, z_n).$$

Then, the *Vapnik-Chervonenkis* (VC) dimension of $\mathcal{A}$, denoted by $V_{\mathcal{A}}$, is the largest integer $k \geq 1$ such that $s(\mathcal{A}, k) = 2^k$. The following important identity [21] relates the shatter coefficient to VC dimension:

$$s(\mathcal{A}, n) = \begin{cases} 2^n & \text{if } n \leq V_{\mathcal{A}} \\ 2\frac{n^{V_{\mathcal{A}}}}{V_{\mathcal{A}}!} & \text{if } n > V_{\mathcal{A}} \end{cases}$$

Then we have $P^n_{X,Y}\left[\sup_{\phi\in\mathcal{A}} |\hat{L}(\phi) - L(\phi)| > \epsilon\right] \leq 8s(\mathcal{A}, n)e^{-n\epsilon^2/32}$. which in turn implies $P^n_{X,Y}\left[L(\hat{\phi}) - \min_{\phi\in H} L(\phi)| > \epsilon\right] \leq 8s(\mathcal{H}, n)e^{-n\epsilon^2/128}$. Thus, given a sample of size $n = \frac{128}{\epsilon^2}\left(\ln s(\mathcal{H}, n) + \ln(8/\delta)\right)$ we have

$$P^n_{X,Y}\left[L(\hat{\phi}) - \min_{\phi\in\mathcal{H}} L(\phi) > \epsilon\right] < \delta,$$

irrespective of the distribution $P_{X,Y}$.

4.2 Isolation Fusers

We consider a family of fuser functions $\mathcal{F} : \{f : \{0,1\}^N \mapsto \{0,1\}\}$ such that the fused output is given by $f[\hat{\phi}_1(X), \hat{\phi}_2(X), \ldots, \hat{\phi}_N(X)]$, denoted by $f(Z)$, where $Z = (\hat{\phi}_1(X), \hat{\phi}_2(X), \ldots, \hat{\phi}_N(X))$. The error probability of the fused system is

$$L_F(f) = \int I_{\{f(Z) \neq Y\}} dP_{X,Y}.$$

Note that Z is a deterministic function of X given the sample. For computational convenience, we utilize the following alternative formula

$$L_F(f) = \int [f(Z) - Y]^2 dP_{X,Y}.$$

Note that $|\mathcal{F}| \leq 2^{2^N}$ since $\mathcal{F}$ consists of at most all Boolean functions on N variables. Consider the function class

$$\mathcal{G} = \{f(\phi_1(X), \phi_2(X), \ldots, \phi_N(X)) : \phi_1 \in \mathcal{H}_1, \phi_2 \in \mathcal{H}_2, \ldots, \phi_N \in \mathcal{H}_N\}.$$

Here $f(\phi_1(.), \phi_2(.), \ldots, \phi_N(.))$ specifies a subset of $\Re^d$, and hence $\mathcal{G}$ specifies a family of sets of $\Re^d$.

The fuser is obtained in two steps: (a) a training set (Z_1, Y_1), (Z_2, Y_2), ..., (Z_n, Y_n), where $Z_i = (\hat{\phi}_1(X_i), \hat{\phi}_2(X_i), \ldots, \hat{\phi}_N(X_i))$, is derived from the classifiers and the original sample, and (b) the fuser is derived by minimizing empirical error over $\mathcal{F}$. Let f^* minimize $L_F(.)$ over $\mathcal{F}$. Consider the *empirical error*

$$\hat{L}_F(f) = \frac{1}{n}\sum_{i=1}^{n}[f(Z_i) - Y_i]^2.$$

Let $\hat{f}$ minimize $\hat{L}_F(.)$ over $\mathcal{F}$.

If one of the classifier is to be chosen, the lowest achievable error is given by $\min_{i=1}^{N} L(\phi_i^*)$. Since the classifiers can be correlated in an arbitrary manner, the empirically best classifier $\hat{\phi}_{\min} = \arg\min_i \hat{L}(\hat{\phi}_i)$ yields the following guarantee

$$P^n_{X,Y}\left[L(\hat{\phi}_{\min}) - \min_{i=1}^{N} L(\phi_i^*) > \epsilon\right] < \delta_1 + \delta_2 + \ldots + \delta_N.$$

The fuser, thus, provides a *better guarantee* if $\delta_F < \delta_1 + \delta_2 + \ldots + \delta_N$ where

$$P^n_{X,Y}\left[L_F(\hat{f}) - \min_{i=1}^{N} L(\phi_i^*) > \epsilon\right] < \delta_F.$$

The fuser class $\mathcal{F}$ satisfies the *isolation property* [17] if it contains the following N functions: for all $i = 1, 2, \ldots, N$ we have $f_i(z_1, z_2, \ldots, z_N) = z_i$. This property is trivially satisfied if $\mathcal{F}$ consists of all Boolean functions of N variables.

Although it is sufficient to include N functions in $\mathcal{F}$ to satisfy this property, in general a richer class performs better in practice [11].

If the fuser class $\mathcal{F}$ satisfies the isolation property, then fuser $\hat{f}$ provides better guarantee than the best classifier under the condition $|\mathcal{F}| \leq \frac{1}{2} \sum\limits_{i=1}^{N} \delta_i e^{\epsilon^2 n/2}$ (see [11] for the proof). A minimal realization of this result can be based on $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$ as per the isolation property. We wish to emphasize that this fusion method can be easily applied without identifying the best classifier, while still ensuring its performance in the fused system. The above condition can also be expressed in terms of the VC dimensions as follows

$$|\mathcal{F}| \leq 4 \sum_{i=1}^{N} \frac{(n)^{V_{\mathcal{H}_i}}}{V_{\mathcal{H}_i}!} e^{-\epsilon^2 63n/128}.$$

by noting that $\delta_i = \frac{8(n)^{V_{\mathcal{H}_i}}}{V_{\mathcal{H}_i}!} e^{-\epsilon^2 n/128}$ for $n > \max(V_{\mathcal{H}_1}, V_{\mathcal{H}_2}, \ldots, V_{\mathcal{H}_N})$ [21].

5 Projective Fusers for Function Estimation

The problem of function estimation based on empirical data arises in a number of disciplines such as statistics, systems theory, and computer science. As a result, there has been a profusion of function estimators, whose performance conditions could be quite involved and beyond the expertise of an average practitioner. Nevertheless, several of these estimators are based on considerable practical and theoretical insights, and it would be most desirable to retain their strengths.

We are required to estimate a function $f : [0,1]^d \mapsto [0,1]$, based on a finite *sample* $(X_1, f(X_1)), (X_2, f(X_2)), \ldots, (X_l, f(X_l))$ where $X_1, X_2, \ldots, X_l$, for $l < \infty$, are iid according to an *unknown* distribution P_X on $[0,1]^d$. For an estimator $\hat{f}$ of f we consider the *expected square error* given by

$$I(\hat{f}) = \int (f(X) - \hat{f}(X))^2 dP_X.$$

We are given N previously computed function estimators (as in [14]) each obtained by using an existing method. The individual estimator $\hat{f}_i$ could be a potential function estimator, radial basis function, k-nearest neighbor estimator, regressogram, kernel estimator, regression tree or another estimator.

Given the estimators $\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_N$, we consider that the *fuser* is a function $f_F : [0,1]^N \mapsto [0,1]$ such that $f_F(X, \hat{f}_1(X), \hat{f}_2(X), \ldots, \hat{f}_N(X))$ is the *fused estimate* of $f(X)$. The *expected* and *empirical errors* of the fuser are respectively given by

$$I(f_F) = \int [f(X) - f_F(X, \hat{f}_1(X), \hat{f}_2(X), \ldots, \hat{f}_N(X))]^2 dP_X$$

$$\hat{I}(f_F) = \frac{1}{l} \sum_{i=1}^{l} [f(X_i) - f_F(X_i, \hat{f}_1(X_i), \hat{f}_2(X_i), \ldots, \hat{f}_N(X_i))]^2.$$

5.1 Class of Projective Fusers

A *projective fuser* [15], f_P, corresponding to a *partition* $P = \{\pi_1, \pi_2, \ldots, \pi_k\}$, $k \leq N$, of input space $[0,1]^d$ of X ($\pi_i \subseteq [0,1]^d$, $\bigcup_{i=1}^{k} \pi_i = [0,1]^d$, and $\pi_i \cap \pi_j = \phi$ for $i \neq j$), assigns to each block π_i to an estimator $\hat{f}_j$ such that

$$f_P(X, \hat{f}_1, \ldots, \hat{f}_N) = \hat{f}_j(X)$$

for all $X \in \pi_i$. For simplicity, we denote $f_P(X, \hat{f}_1, \ldots, \hat{f}_N)$ by $f_P(X)$. An *optimal projective fuser*, denoted by f_{P^*}, minimizes $I(.)$ over all projective fusers corresponding to all partitions of $[0,1]^d$ and assignments of blocks to estimators.

We define the *error curve* of the estimator $\hat{f}$ for f as $\mathcal{E}(X, \hat{f}) = (f(X) - \hat{f}(X))^2$. The projective fuser based on *lower envelope of error curves* is defined by

$$f_{LE}(X, \hat{f}_1, \ldots, \hat{f}_N) = \hat{f}_{i_{LE}(X)}(X)$$

where $i_{LE}(X) = \arg \min_{i=1,2,\ldots,N} \mathcal{E}(X, \hat{f}_i)$. In other words, $f_{LE}(X, \hat{f}_1, \ldots, \hat{f}_N)$ simply outputs the estimator with the lowest error at X. Thus, we have $\mathcal{E}(X, f_{LE}) = \min_{i=1}^{N} \mathcal{E}(X, \hat{f}_i)$, or equivalently the error curve of f_{LE} is the lower envelope with respect to X of the set of error curves $\{\mathcal{E}(X, \hat{f}_1), \ldots, \mathcal{E}(X, \hat{f}_N)\}$.

5.2 Nearest Neighbor Projective Fuser

We partition the space of X into Voronoi regions $V(X_1), V(X_2), \ldots, V(X_l)$ such that

$$V(X_j) = \{X : \parallel X - X_j \parallel < \parallel X - X_k \parallel \text{ for all } k = 1, 2, \ldots, l; k \neq j\}$$

where $\parallel . \parallel$ is the Euclidean metric. The points equidistant from more than one sample point are arbitrarily assigned to one of the regions. We assume that all X_i's are distinct without the loss of generality. $V(X_j)$ is simply the set of all points that are at least as close to X_j as to any other X_k. Let $NN(X) = k$ such that $X \in V(X_k)$ for some k, which is the Voronoi cell that X belongs to. For the cell $V(X_{NN(X)})$ that contains X, we identity the estimator that achieves the lowest empirical error at the sample point $X_{NN(X)}$ by defining the *estimator index* of X as follows

$$i_{NN}(X) = \arg \min_{i=1,2,\ldots,N} [f(X_{NN(X)}) - \hat{f}_i(X_{NN(X)})]^2.$$

That is, $i_{NN}(X)$ is the index of the estimator that achieves least empirical error at the sample point $X_{NN(X)}$ nearest to X. Then the *nearest neighbor projective fuser* [19] is defined as

$$\hat{f}_{NN}(X, \hat{f}_1(X), \ldots, \hat{f}_N(X)) = \hat{f}_{i_{NN}(X)}(X).$$

Despite the notational complexity, the idea of $\hat{f}_{NN}$ is quite simple: $\hat{f}_{NN}(X)$ is $\hat{f}_i(X)$ that achieves least empirical error at the nearest sample point to X.

5.3 Sample-Based Projective Fusers

The computation of f_{LE} in general requires a complete knowledge of the distribution P_X. To address the case where such knowledge is not available, a method was proposed in [15] that utilizes regression estimation methods to compute an estimator $\hat{\mathcal{E}}(X, \hat{f}_i)$ of $\mathcal{E}(X, \hat{f}_i)$, and utilizes the lower envelope of these estimators in the computation of fuser. We now briefly outline the basic approach using the cubic partitions with data-dependent offsets for $d = 1$. For a sequence $\{h_l\}$ of positive numbers, consider the partition of $\Re$ given by $\theta_l = \{[(r-1)h_l, rh_l)|r \in \mathcal{Z}\}$. Let $\psi_l[X]$ denote the unique cell of θ_l that contains X. Then, the estimator of $\mathcal{E}(X, \hat{f}_i)$ is given by

$$\hat{\mathcal{E}}(X, \hat{f}_i) = \frac{\sum\limits_{j=1}^{l} (X_j - \hat{f}_i(X_j))^2 1_{\psi_l[X]}(X_j)}{\sum\limits_{j=1}^{n} 1_{\psi_l[X]}(X_j)}.$$

In other words, the estimator simply computes the mean of the error of $\hat{f}_i$ within the cell of θ_l that contains X. Consider the conditions: (i) $((X - f(Y))^2 < K$ for some $K > 0$; (ii) $\lim\limits_{l \to \infty} h_l \to 0$; and (iii) $nh_l \to \infty$ as $l \to \infty$. Then, we have $\int |\mathcal{E}(X, \hat{f}_i) - \hat{\mathcal{E}}(X, \hat{f}_i)|^2 dP_X \to 0$ with probability 1 [8], regardless of the distribution P_X. The fuser $\hat{f}_{LE}$ is computed using $\hat{\mathcal{E}}(X, \hat{f}_i)$ in place of $\mathcal{E}(X, \hat{f}_i)$. The strong consistency of $\hat{f}_{LE}$ method is shown under the boundedness of $(X - f(X))^2$, namely $I(\hat{f}_{LE}) \to I(f_{LE})$ as $l \to \infty$ with probability 1 for any distribution P_X [15]. This result specifies the performance of $\hat{f}_{LE}$ for sample sizes approaching infinity and does not tell much when sample sizes are finite. The implementation of $\hat{f}_{LE}$ itself is tricky in that the choice of h_l is not evident if finite-sample performance is needed.

The individual function estimators $\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_N$ could be quite varied, but several of them satisfy certain smoothness or non-smoothness conditions. For any function $g : [-A, A]^d \mapsto \Re$, let $\parallel g(r) \parallel_\infty = \sup\limits_{r \in [-A,A]^d} |g(r)|$. A function $g(y) : [-A, A]^d \mapsto \Re^a$ is *Lipschitz* with constant k_g if for all $y_1, y_2 \in [-A, A]^d$, we have $\parallel g(y_1) - g(y_2) \parallel_\infty \leq k_g \parallel y_1 - y_2 \parallel_\infty$. The examples of smooth function estimators include potential functions, sigmoid neural networks, smooth kernel estimates, radial basis functions, linear and polynomial estimators.

Several function estimators are not Lipschitz with popular examples including nearest neighbor and Nadaraya-Watson estimators. To address such cases we consider the class of functions with bounded variation, which allows for discontinuities and includes Lipschitz functions as a subclass. Consider one-dimensional function $h : [-A, A] \mapsto \Re$. For $A < \infty$, a set of points $P = \{y_0, y_1, \ldots, y_n\}$ such that $-A = y_0 < y_1 < \ldots < y_n = A$ is called a *partition* of $[-A, A]$. The collection of all possible partitions of $[-A, A]$ is denoted by $\mathcal{P}[-A, A]$. A function $g : [-A, A] \mapsto \Re$ is of *bounded variation*, if there exists the total variation M such that for any partition $P = \{y_0, y_1, \ldots, y_n\}$, we have $\sum\limits_{k=1}^{n} |g(y_k) - g(y_{k-1})| \leq M$. A multivariate function $g : [-A, A]^d \mapsto \Re$ is of bounded variation if it is so in each of its input variable for every value of the other input variables. The following are useful facts about the functions of bounded variation: (i) not all continuous functions are of bounded variation, e.g. $g(y) = y\cos(\pi/(2y))$ for $y \neq 0$ and $g(0) = 0$; (ii) differentiable functions on compact domains are of bounded variation; and (iii) absolutely continuous functions, which include Lipschitz functions, are of bounded variation.

The function estimators such as k-nearest neighbor, Haar wavelet estimators, regression tree, regressogram and Nadaraya-Watson estimator (which all could involve discrete jumps) satisfy the bounded variation property. Since Lipschitz estimators over compact domains also have bounded variation, the latter is a fairly general property satisfied by most of the widely-used estimators.

We consider that the function estimators $\hat{f}_1, \ldots, \hat{f}_N$ are of bounded variation. Let each function estimator $\hat{f}_i$ be of total variation V_i. For $V = \sum_{i=1}^{N} V_i$, it is shown in [19] that $P\left[I(\hat{f}_{NN}) - I(f_{LE}) > \epsilon\right] < \delta$ for sample size

$$\frac{256}{\epsilon^2}\left[18\left(1 + \frac{128V}{\epsilon}\right)\ln^2(128/\epsilon) + \ln(16/\delta)\right].$$

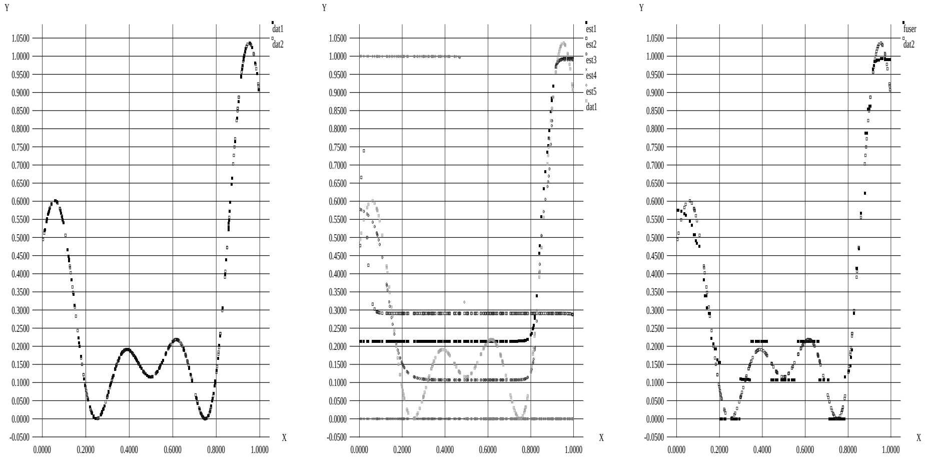
Furthermore, $I(\hat{f}_{NN}) \rightarrow I(f_{LE})$ as $l \rightarrow \infty$. This result establishes the analytical viability of $\hat{f}_{NN}$ for finite samples. While the sample size estimate is not necessarily within practical limits, the overall result itself is stronger than the asymptotic consistency.

5.4 Computational Example

We consider the problem of estimating

$$f(X) = 0.02(12 + 3X + 7.2x^2)(1.0 + \cos(4\pi X))(1.0 + 0.8\sin(3\pi X/7))$$

based on a sample. Two samples each of size 200 (Fig. 1(a)) are used in training the neural networks and fuser. Five feedforward neural networks are trained using the backpropagation algorithm with different starting weights and different learning rates as shown in Fig. 1(b). The performance of the estimators and fuser is measured by the empirical error on the sample. The estimator 1 approximated well only in the vicinity of $X = 1$, whereas estimator 2 is close to the function in the vicinity of $X = 0$. Estimator 3 provided a good approximation at both ends of the interval $[0, 1]$ and is the best of the estimators. However, this estimator is insensitive to the variations of $f(X)$ in the middle of the interval $[0, 1]$. Estimator 4 performs the worst staying close to 0 for entire $[0, 1]$. The performance of $\hat{f}_{NN}$ is shown in Figure 1(c) which is uniformly as good as any of the estimators across the entire interval. The best estimator 3 is used by the fuser for the most of the interval $[0, 1]$ except in the middle. It is interesting to note that the worst estimator, namely, estimator 4, is used in the lowest portions of $f(X)$, and indeed is responsible for the better performance achieved by the fuser.



(a) (b) (c)

Fig. 1. *Nearest neighbor projective fuser for function estimators.*

6 Conclusions

A generic sensor fusion problem is formulated for sensors whose measurements are subject to unknown probability distributions. A brief overview of fuser design methods is presented with a focus on finite sample performance guarantees. The classes of isolation and projective fusers are described for the special cases of classification and function estimation. Similar concepts have been studied in multiple classifier systems [3, 23]. The methods described in this paper have been applied in practice for combining ultrasonic and infrared sensor measurements for robot navigation, prediction of embrittlement levels in light water reactors, combining sensor readings of well data in methane hydrate explorations, and combining radar measurements for target detection.

Several open problems remain in the generic sensor fusion problem as well as in classification and function estimation. Often the sample bounds are too large to be practical, and the performance equations do not provide uniform precision in that the sensor with best bound is not necessarily the best. It would be interesting to develop principles that bridge the gap between performance bounds and actual performance. Also there has been a profusion of fusion concepts of significant diversity, and it would be interesting to identify unifying principles behind these developments.

Acknowledgments

This research is sponsored by the Engineering Research Program of the Office of Science, U.S. Department of Energy, under Contract No. DE-AC05-00OR22725 with UT-Battelle, LLC.

References

1. C. K. Chow. Statistical independence and threshold functions. *IEEE Trans. Electronic Computers*, EC-16:66–68, 1965.
2. L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, New York, 1996.
3. G. Giacinto and F. Roli. Dynamic classifier selection based on multiple classifier behavior. *Pattern Recognition*, 34:1879–1881, 2001.
4. B. Grofman and G. Owen, editors. *Information Pooling and Group Decision Making*. Jai Press Inc., Greenwich, Connecticut, 1986.
5. J. Kittler and F. Roli, editors. *Multiple Classifier Systems*, volume 1857. Springer-Verlag, Berlin, 2000.
6. V. Kurkova. Kolmogorov's theorem and multilayer neural networks. *Neural Networks*, 5:501–506, 1992.
7. R. N. Madan and N. S. V. Rao. Guest editorial on information/decision fusion with engineering applications. *Journal of Franklin Institute*, 336B(2), 1999. 199-204.
8. A. Nobel. Histogram regression estimation using data-dependent partitions. *Annals of Statistics*, 24(3):1084–1105, 1996.
9. N. S. V. Rao. Distributed decision fusion using empirical estimation. *IEEE Transactions on Aerospace and Electronic Systems*, 33(4):1106–1114, 1996.
10. N. S. V. Rao. Fusion methods in multiple sensor systems using feedforward neural networks. *Intelligent Automation and Soft Computing*, 5(1):21–30, 1998.
11. N. S. V. Rao. To fuse or not to fuse: Fuser versus best classifier. In *SPIE Conference on Sensor Fusion: Architectures, Algorithms, and Applications II*, pages 25–34. 1998.
12. N. S. V. Rao. Vector space methods for sensor fusion problems. *Optical Engineering*, 37(2):499–504, 1998.
13. N. S. V. Rao. Multiple sensor fusion under unknown distributions. *Journal of Franklin Institute*, 336(2):285–299, 1999.
14. N. S. V. Rao. On optimal projective fusers for function estimators. In *Second International Conference on Information Fusion*, pages 296–301. 1999.
15. N. S. V. Rao. Projective method for generic sensor fusion problem. In *IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 1–6. 1999.
16. N. S. V. Rao. Finite sample performance guarantees of fusers for function estimators. *Information Fusion*, 1(1):35–44, 2000.
17. N. S. V. Rao. On fusers that perform better than best sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):904–909, 2001.
18. N. S. V. Rao. Multisensor fusion under unknown distributions: Finite sample performance guarantees. In A. K. Hyder, editor, *Multisensor Fusion*. Kluwer Academic Pub., 2002.
19. N. S. V. Rao. Nearest neighbor projective fuser for fucntion estimation. In *Proceedings of International Conference on Information Fusion*, 2002.
20. N. S. V. Rao and S. S. Iyengar. Distributed decision fusion under unknown distributions. *Optical Engineering*, 35(3):617–624, 1996.
21. V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.
22. P. K. Varshney. *Distributed Detection and Data Fusion*. Springer-Verlag, 1997.
23. T. Windeatt and F. Roli, editors. *Multiple Classifier Systems*. Springer-Verlag, Berlin, 2003.